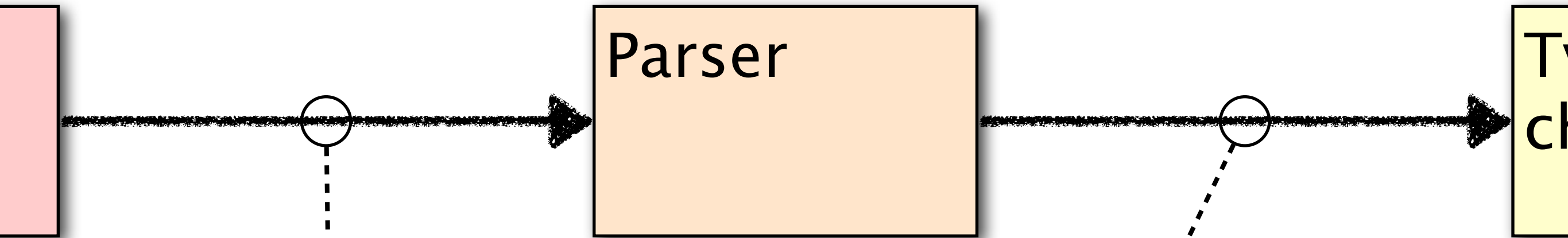


Character stream:

```
while x <> y do (* Inv:  
gcd(x, y) = gcd(a, b) *) if  
x > y then x := x - y  
else y := y - x end end
```

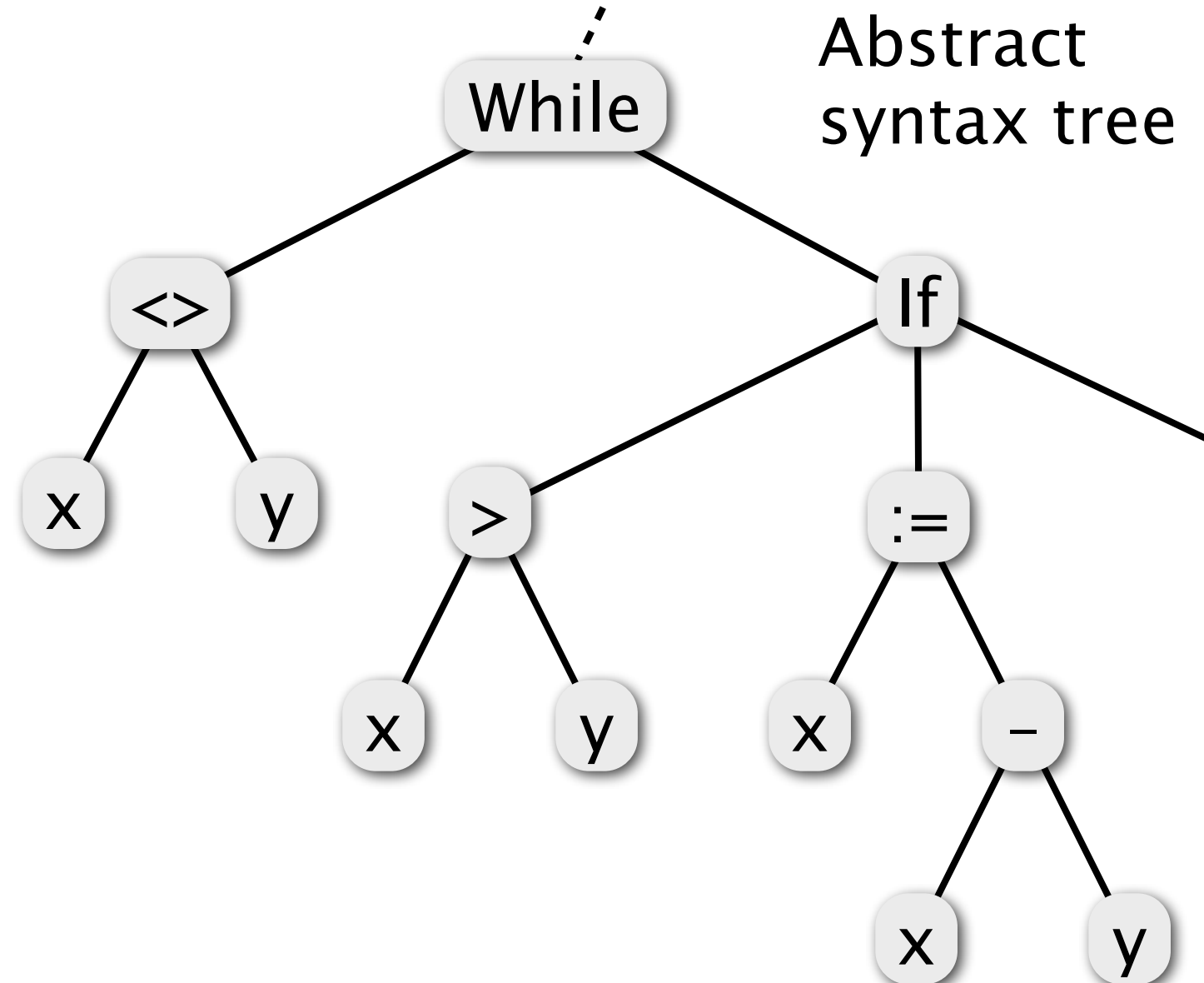
Token Stream:

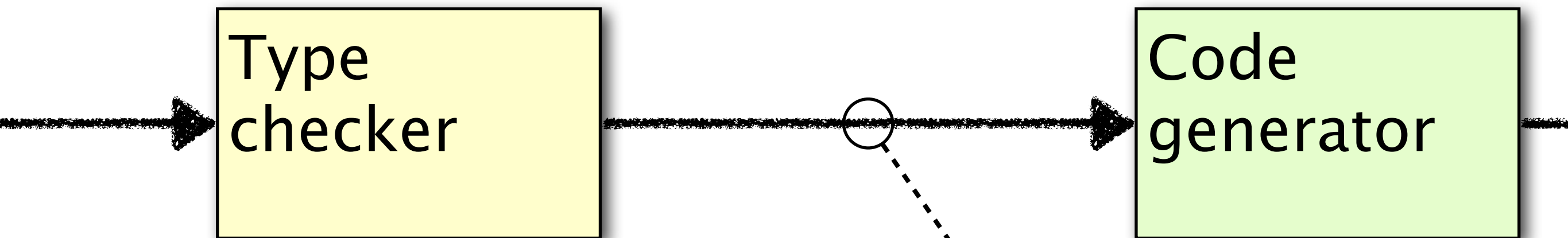
```
WHILE, IDENT "x", RELOP  
NEQ, IDENT "y", DO, IF,  
IDENT "x", RELOP GT,  
IDENT "y", THEN, IDENT "x",  
ASSIGN, IDENT "x", MINUS,  
IDENT "y", ELSE, IDENT "y",  
ASSIGN, IDENT "y", MINUS,  
IDENT "x", END, END
```



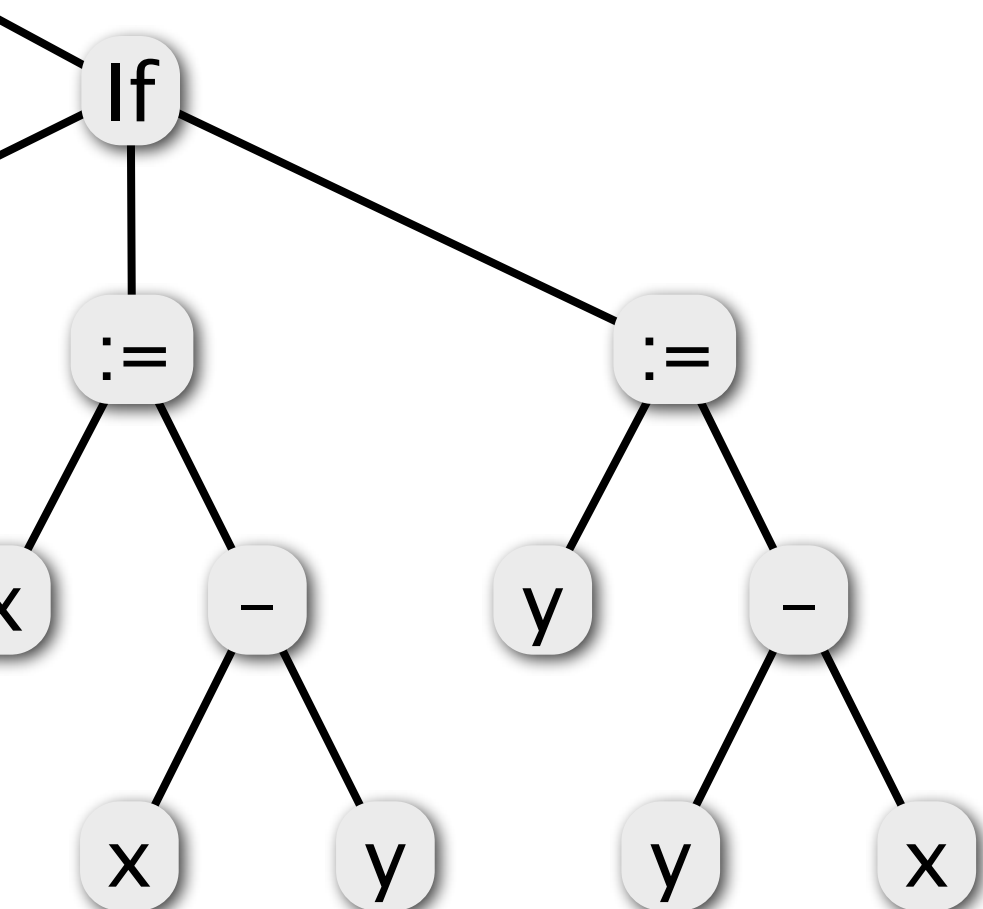
Token Stream:

*WHILE, IDENT "x", RELOP  
NEQ, IDENT "y", DO, IF,  
IDENT "x", RELOP GT,  
IDENT "y", THEN, IDENT "x",  
ASSIGN, IDENT "x", MINUS,  
IDENT "y", ELSE, IDENT "y",  
ASSIGN, IDENT "y", MINUS,  
IDENT "x", END, END*

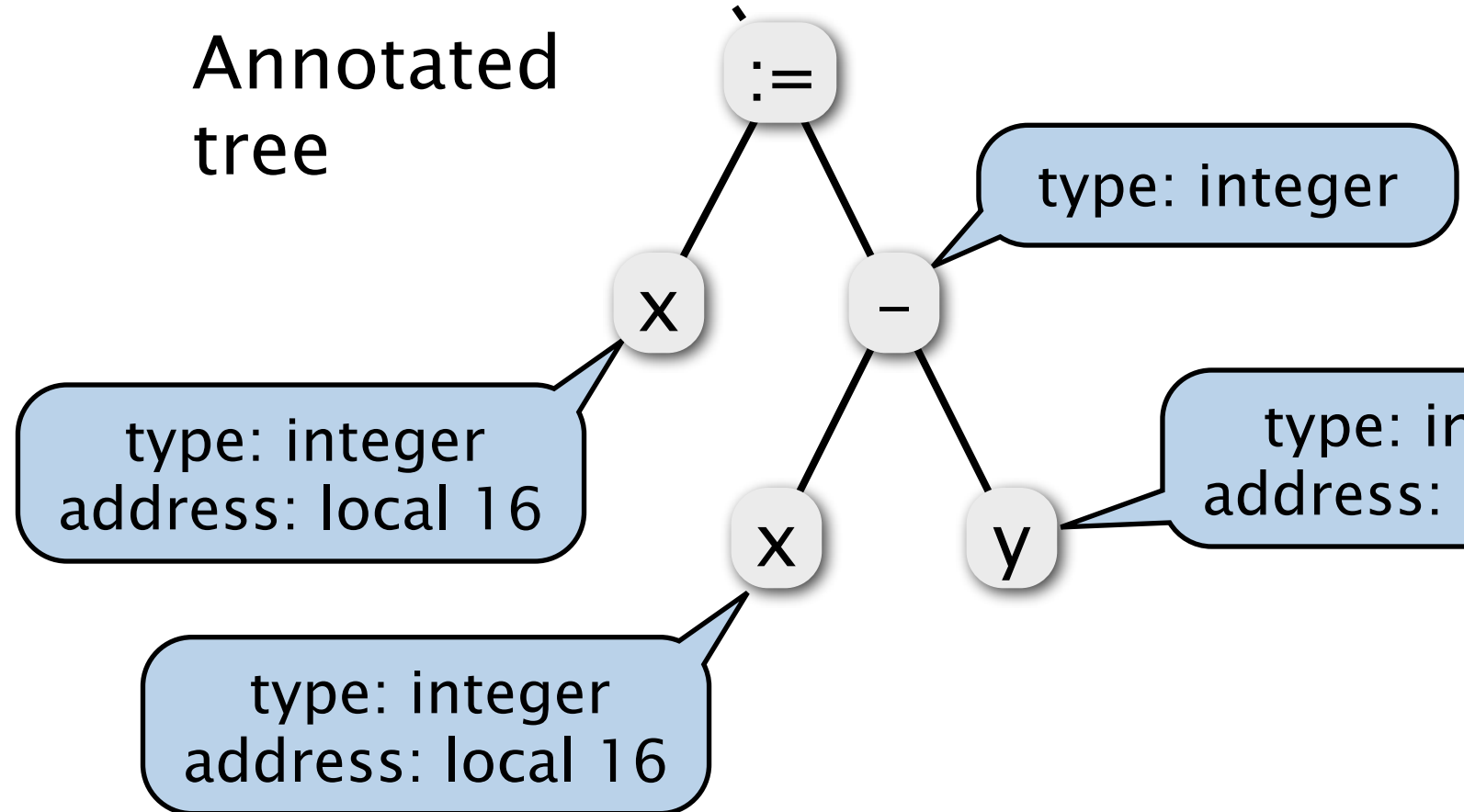


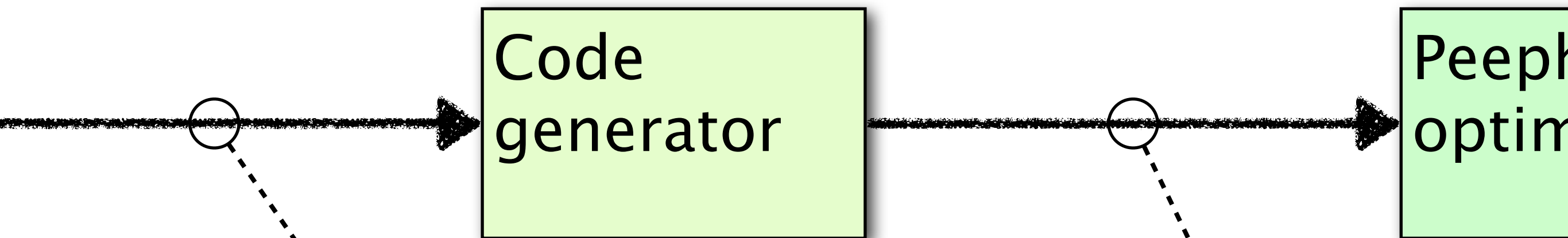


Abstract  
syntax tree

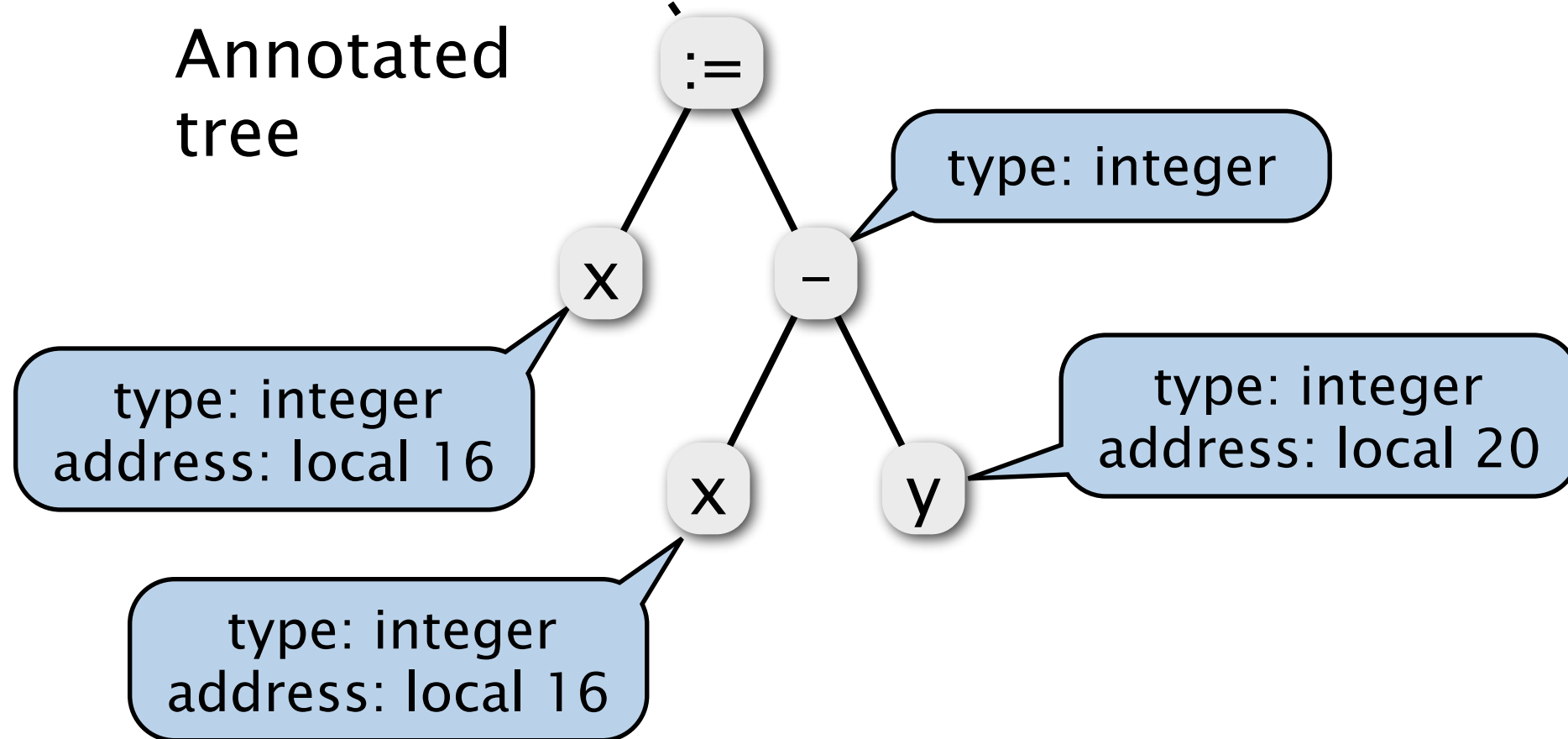


Annotated  
tree



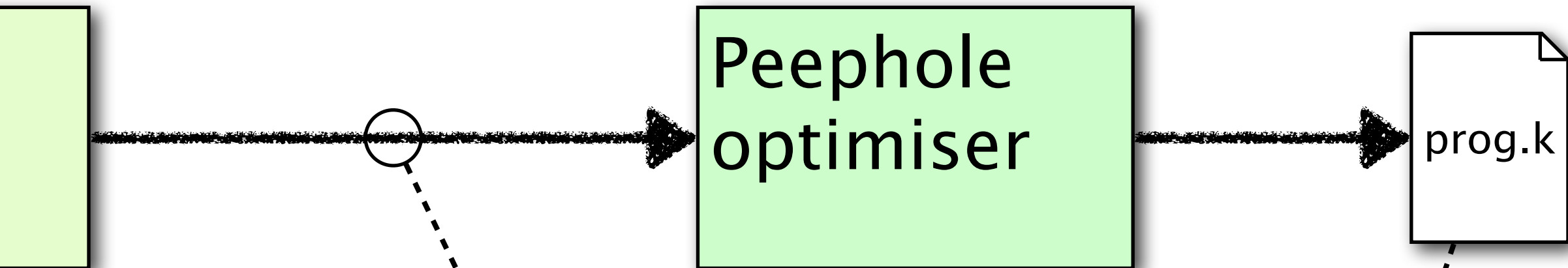


Annotated tree



Postfix code:

```
LOCAL 16  
LOADW  
LOCAL 20  
LOADW  
MINUS  
LOCAL 16  
STOREW
```



integer

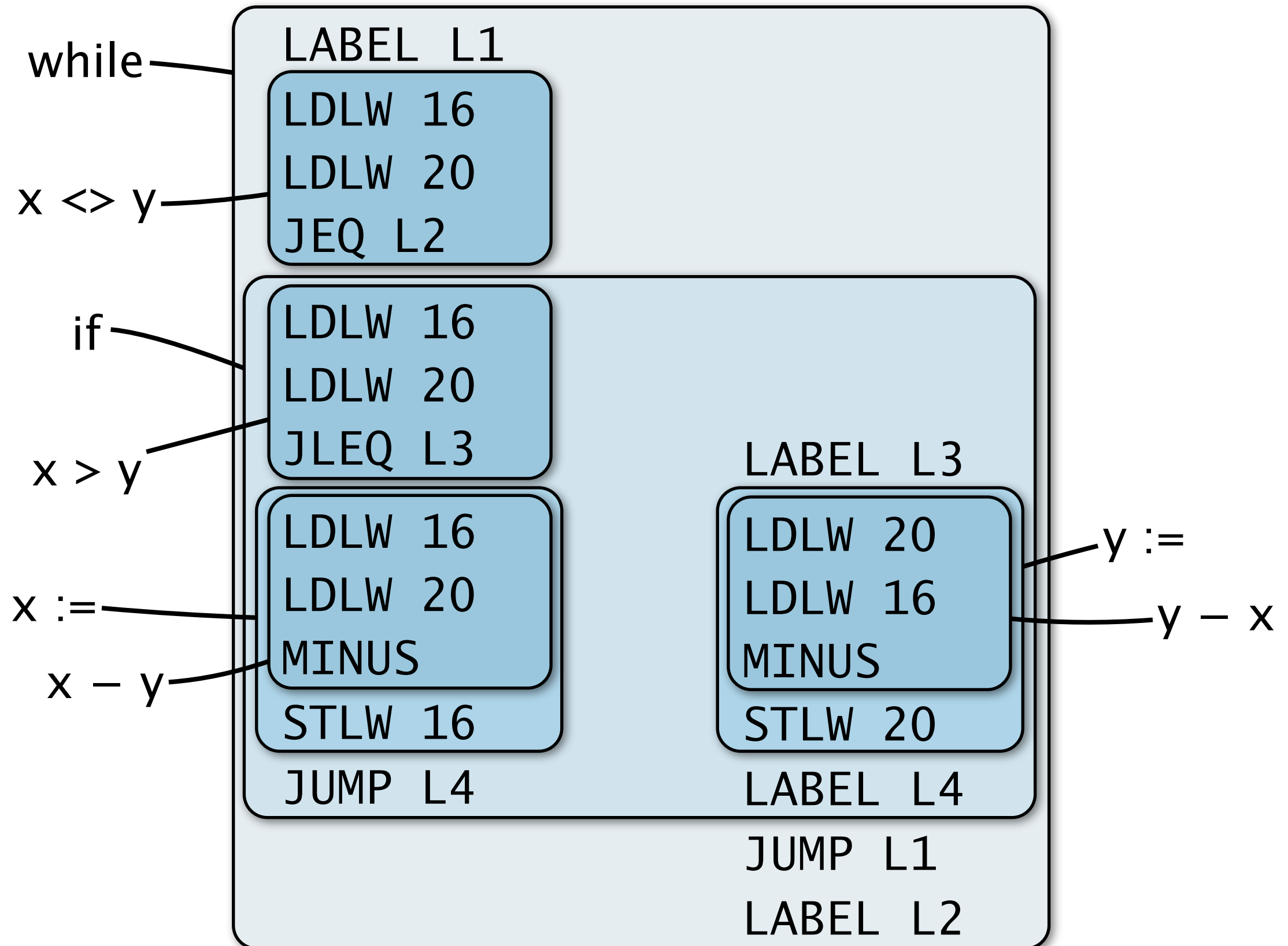
type: integer  
address: local 20

Postfix code:

LOCAL 16  
LOADW  
LOCAL 20  
LOADW  
MINUS  
LOCAL 16  
STOREW

Optimised code:

LDLW 16  
LDLW 20  
MINUS  
STLW 16



LABEL L1

LDLW 16

LDLW 20

JEQ L2

LDLW 16

LDLW 20

JLEQ L3

LDLW 16

LDLW 20

MINUS

STLW 16

JUMP ~~L4~~ L1

LABEL L3

LDLW 20

LDLW 16

MINUS

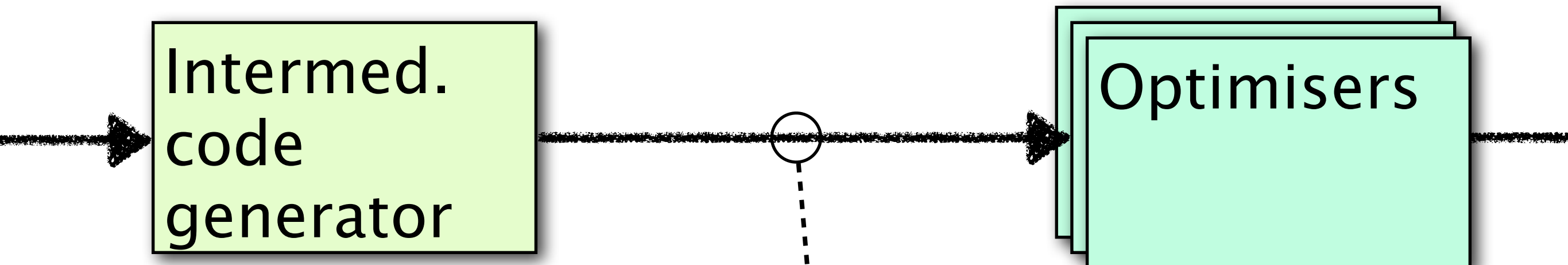
STLW 20

~~LABEL L4~~

JUMP L1

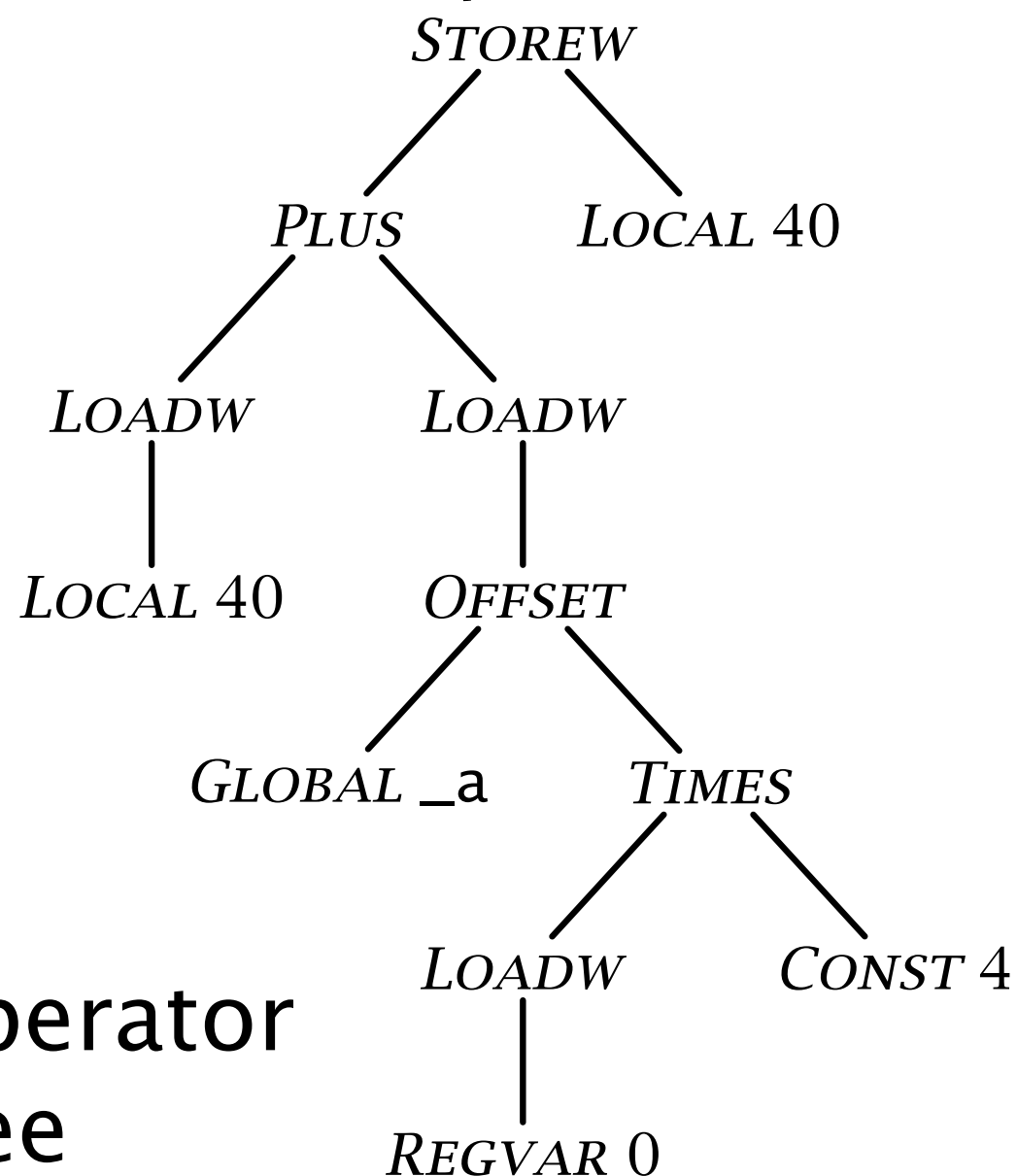
LABEL L2



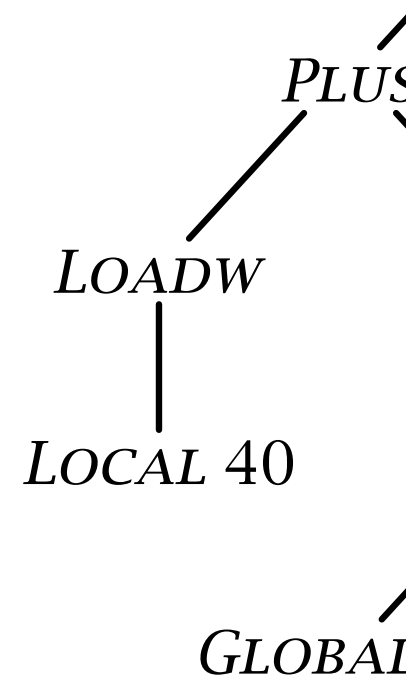


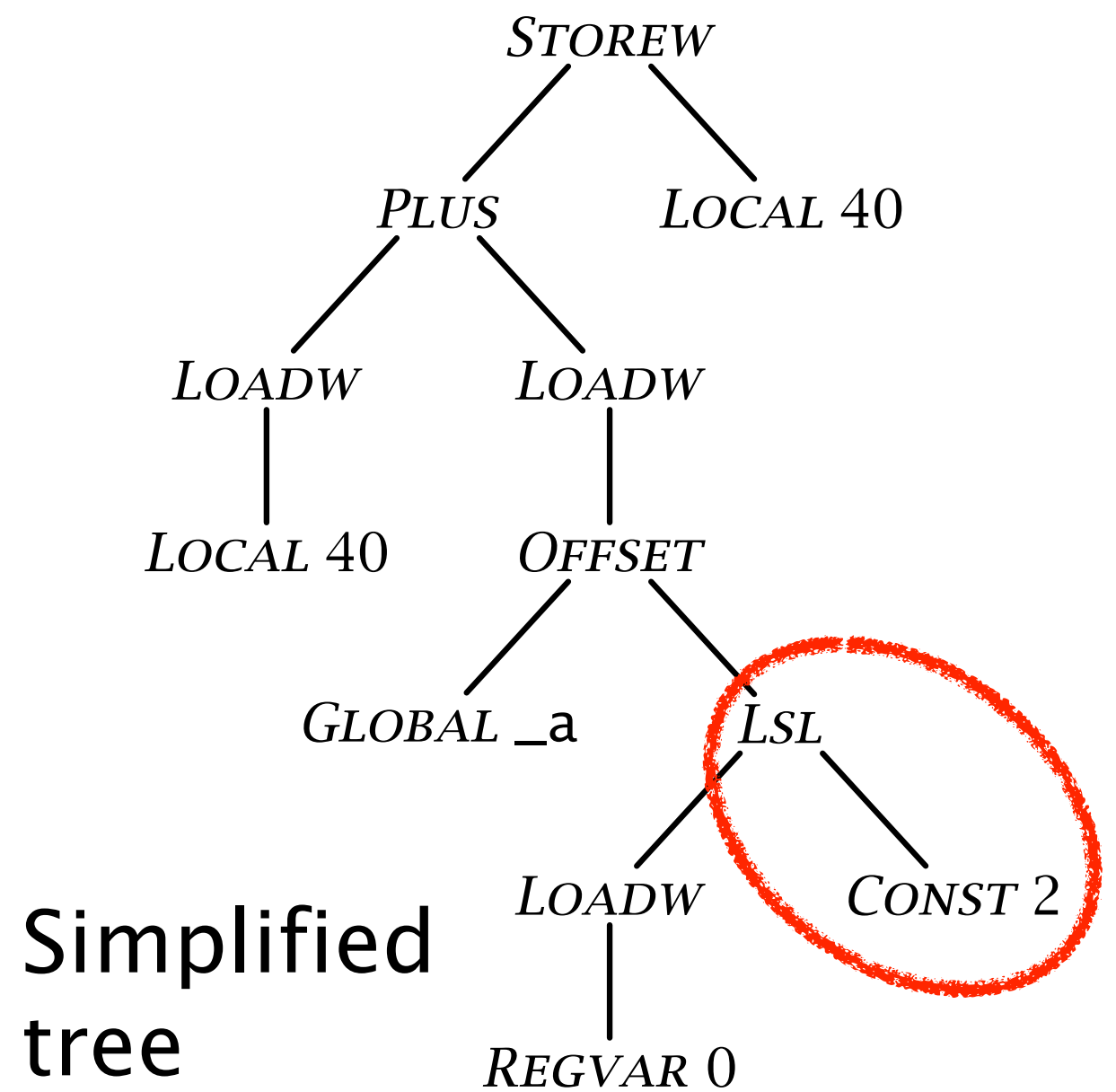
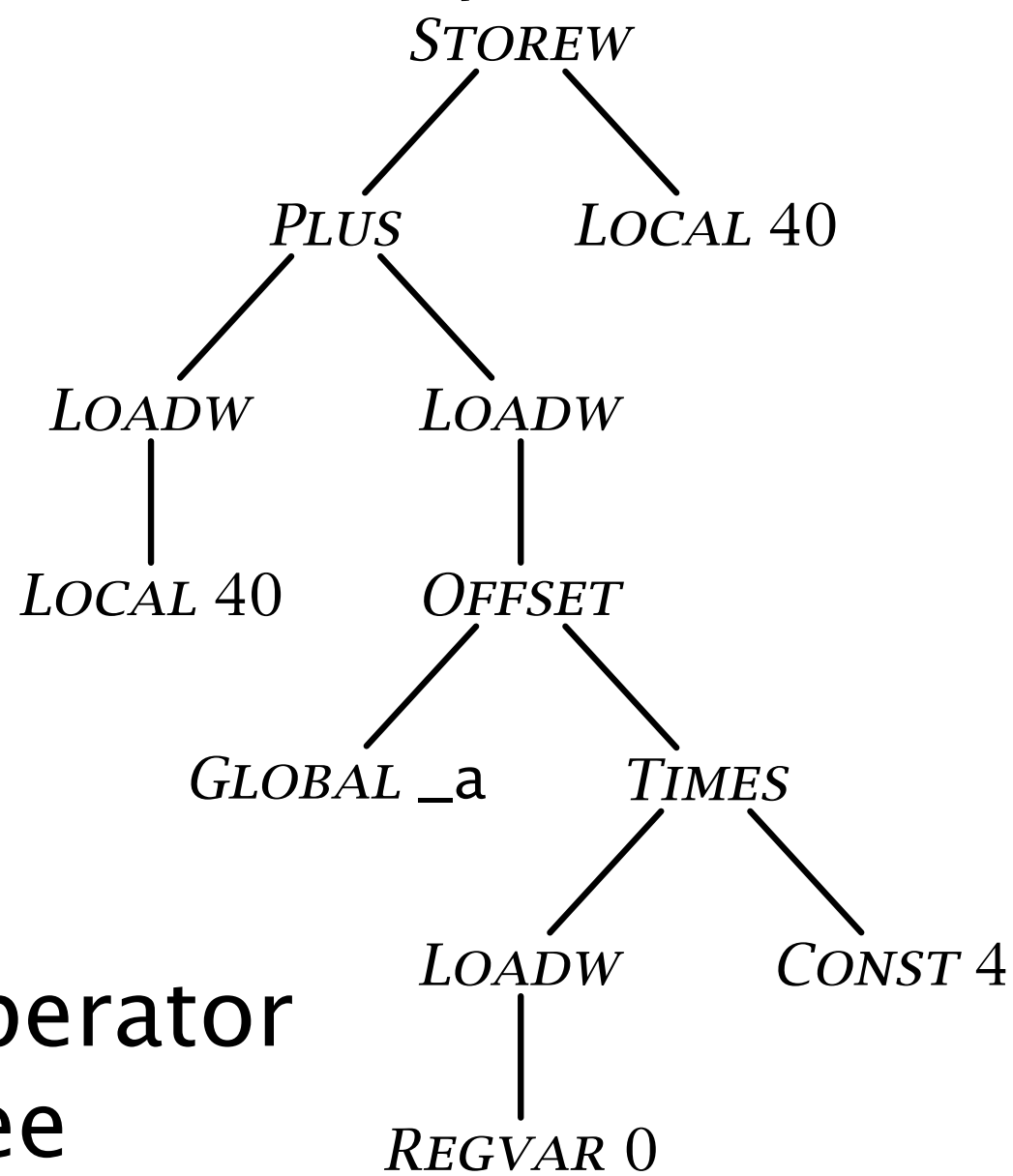
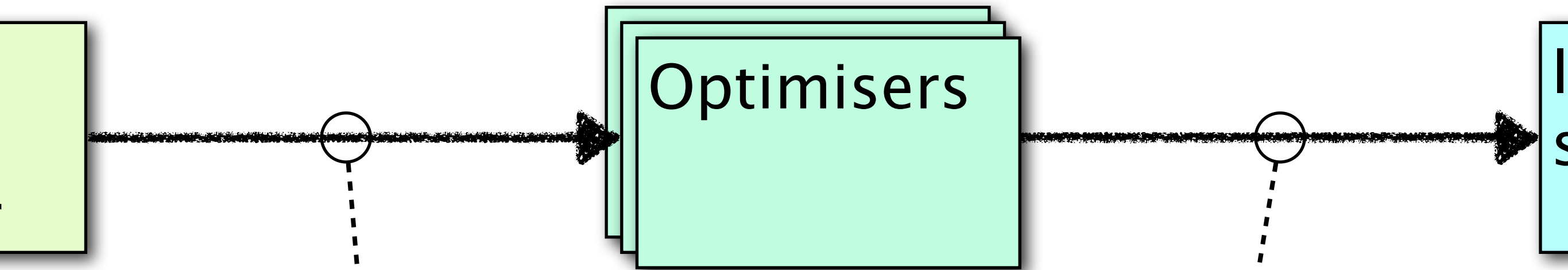
**s := s + a[i]**

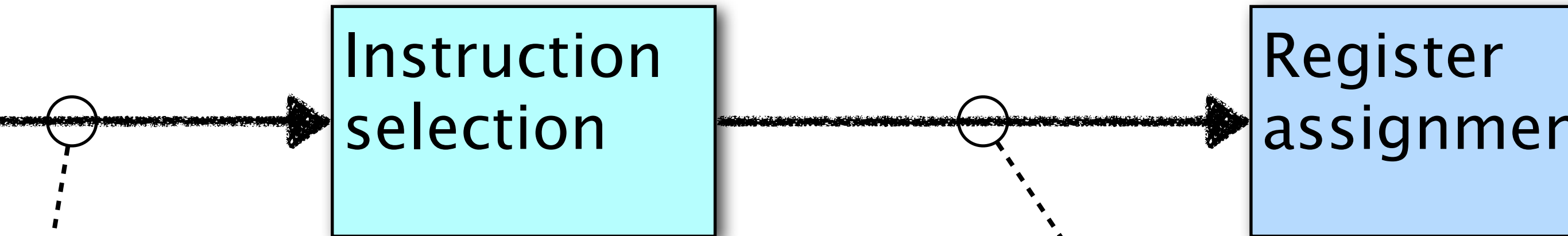
Operator  
tree



Simplified  
tree







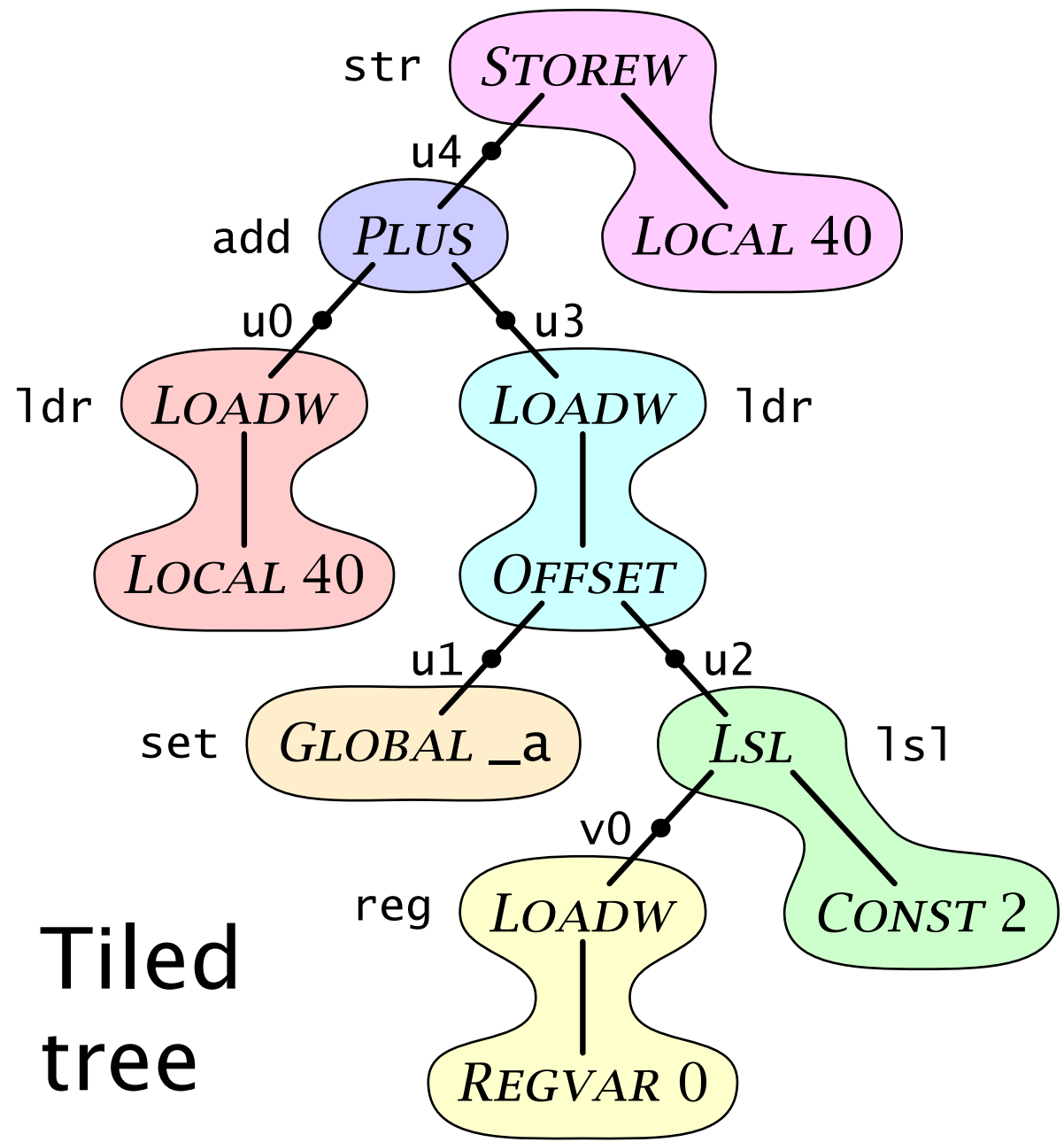
REW  
LOCAL 40

DW

SET

DW

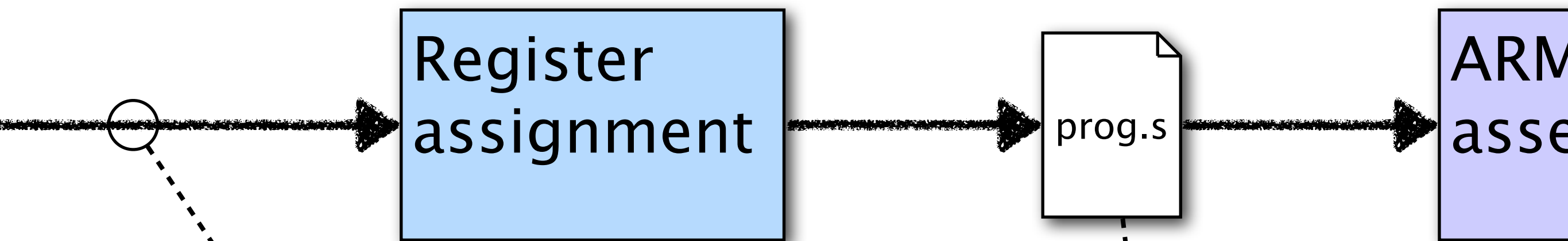
AR 0



Tiled  
tree

Code templates:

```
ldr u0, [fp, #40]
set u1, _a
lsl u2, v0, #2
ldr u3, [u1, u2]
add u4, u0, u3
str u4, [fp, #40]
```



L 40

dr

L

lsl

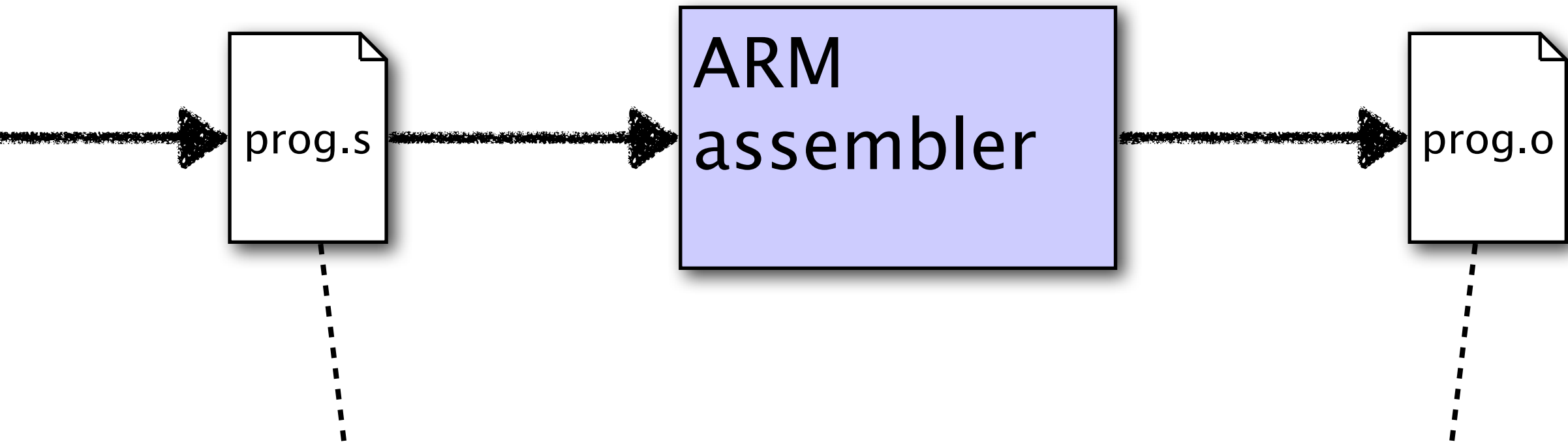
CONST 2

Code templates:

```
ldr u0, [fp, #40]
set u1, _a
lsl u2, v0, #2
ldr u3, [u1, u2]
add u4, u0, u3
str u4, [fp, #40]
```

Assembly code:

```
ldr r0, [fp, #40]
set r1, _a
lsl r2, r4, #2
ldr r1, [r1, r2]
add r0, r0, r1
str r0, [fp, #40]
```



Assembly code:

```
ldr r0, [fp, #40]
set r1, _a
lsl r2, r4, #2
ldr r1, [r1, r2]
add r0, r0, r1
str r0, [fp, #40]
```

Machine code:

```
e59b0028
e59f100c
e1a02104
e7911002
e0800001
e58b0028
```

```

i := 0; s := 0
while i < n do
    if a[i] > 0 then
        s := s + a[i]
    end;
    i := i+1
end

```

```

mov r5, 0
mov r4, 0
.L2: ldr r0, [fp, #40]
     cmp r5, r0
     bge .L4
     set r0, _a
     ldr r6, [r0, r5, LSL #2]
     cmp r6, #0
     ble .L7
     add r4, r4, r6
.L7: add r5, r5, 1
     b .L2
.L4:

```